

Software Design Document

Cloud Database Migration & LACPD New Hire Enrollment Web Application

Version **1.4** approved

Prepared by **Chen Albert, Simon Mai, Shahram Mehri Kalantari, Wilfredo Paz, Michael Loria, Ryan Lee, Ismael Valenzuela, Joshua Perez, Norberto Rosales, Fabio Quintana**

Law Offices of Los Angeles County Public Defender & Cal State LA

13 May 2022

Table of Contents.....	<pg 2-3>
Revision History.....	<pg 4>
1. Introduction.....	<pg 5>
1.1. Purpose.....	<pg 5>
1.2. Document Conventions.....	<pg 5>
1.3. Intended Audience and Reading Suggestions.....	<pg 5>
1.4. System Overview.....	<pg 5>
2. Design Considerations.....	<pg 6>
2.1. Assumptions and dependencies.....	<pg 6>
2.2. General Constraints.....	<pg 6>
2.3. Goals and Guidelines.....	<pg 6>
2.4. Development Methods.....	<pg 7>
3. Architectural Strategies.....	<pg 8>
4. System Architecture.....	<pg 9-10>
5. Policies and Tactics.....	<pg 11>
5.1. Specific Products Used.....	<pg 11>
5.2. Requirements traceability.....	<pg 11>
5.3. Testing the software.....	<pg 11>
5.4. Maintaining the Software.....	<pg 11>
5.5. Interfaces for end-users.....	<pg 11>
6. Detailed System Design.....	<pg 12>
6.1 Angular Single Page Application	<pg 12>
6.1.1 Responsibilities.....	<pg 12>
6.1.2 Constraints.....	<pg 12>
6.1.3 Composition.....	<pg 12>
6.1.4 Uses/Interactions.....	<pg 12>
6.1.5 Resources.....	<pg 12>
6.1.6 Interface/Exports.....	<pg 12>
6.2 Power Apps Applications.....	<pg 13>
6.2.1 Responsibilities.....	<pg 13>
6.2.2 Constraints.....	<pg 13>
6.2.3 Composition.....	<pg 13>
6.2.4 Uses/Interactions.....	<pg 13>
6.2.5 Resources.....	<pg 13>
6.2.6 Interface/Exports.....	<pg 13>
7. Detailed Lower level Component Design.....	
7.1 SQL Database.....	<pg 14>
7.1.1 Classification.....	<pg 14>
7.1.2 Processing Narrative(PSPEC).....	<pg 14>
7.1.3 Interface Description.....	<pg 14>
7.1.4 Processing Detail.....	<pg 14>
7.1.4.1 Design Class Hierarchy.....	<pg 14>
7.1.4.2 Restrictions/Limitations.....	<pg 14>
7.1.4.3 Performance Issues.....	<pg 14>
7.1.4.4 Design Constraints.....	<pg 14>
7.1.4.5 Processing Detail For Each Operation.....	<pg 14>

8.	User Interface	
8.1.	Overview of User Interface.....	<pg 15>
8.2.	Screen Frameworks or Images.....	<pg 15>
8.3.	User Interface Flow Model.....	<pg 15>
9.	Database Design.....	<pg 17>
10.	Requirements Validation and Verification.....	<pg 20>
11.	Glossary.....	<pg 22>
12.	References.....	<pg 23>

Revision History

Name	Date	Reason For Changes	Version
Albert Chen	11/15/21	initial creation	1.0
Albert Chen	11/29/21	filled in 1-7	1.1
Simon Mai	4/23/22	Updated and added information	1.2
Michael Loria	4/26/22	Power Apps information and document structure	1.3
Norberto Gomez Rosales	4/28/22	Updating and added information	1.4

1. Introduction

1.1 Purpose

The purpose of this document is to:

- a) Define the design for the Public Defender's Enrollment Application so that the reader gains a better understanding of the technologies and techniques that are used in the implementation of a web application that handles online LACPD onboarding process.
- b) Define the tools used for the migration of PowerApps data to the Azure SQL database.

The SDD describes only part of the system and provides the information necessary for the system and the software to be assembled.

1.2 Document Conventions

N/A

1.3 Intended Audience and Reading Suggestions

This document is intended for project managers, users, testers, and developers who want to implement or wish to make future changes to design. The rest of this SDD will contain Design Considerations, Architectural Strategies, System Architecture, Policies and Tactics, Name of Component Module, Detailed Lower Level Component Design, Database Design, Overview of User Interface, Requirements, Glossary, and References.

1.4 System Overview

Power Apps - The software system's design includes the integration of Microsoft Power Apps to a cloud database. The overall system will be two applications of Power Apps PDGO and Helpdesk. The two applications will have their data migrated to AzureSQL Database.

Enrollment Application - The front-end of the Enrollment Application consists of a user-friendly UI that allows users to easily fill in the PDF documents without entering repeating data fields and either save their work or submit it for the signing process. Backend maps the data fields from the frontend to the PDF form field values to make the forms for approval and signature. Documents will be created and sent to approvers for signing, from there the document will be sent to BOX for storage.

2. Design Considerations

2.1 Assumptions and Dependencies

Power Apps

- VPN to access LACPD Azure SQL database
- Office 365 Power Apps Development Environment Permissions

Enrollment Application

- A server will be needed to host the application
- SQL Database for storing the generated documents.
- Adobe Sign REST API is used to generate and fill out PDFs.

2.2 General Constraints

Power Apps

- Full developer credentials provided by LACPD

Enrollment Application

- Adobe Sign API Tokens
- Requires a web server to be able to use and maintain the application.
- Designed for computers with operating systems like Windows or Mac.

2.3 Goals and Guidelines

Power Apps

- The end product should have a database connected and managed by Azure services
- Product should be accessible for desktop and mobile use

Enrollment Application

- The final product should speed up the process of the form filling process.
- The product should have a user-friendly UI that is easy to follow.
- The product should allow a user to easily create an account for the process of the form filling process.
- The product should feel work, look, and feel like a professional workplace product.

2.4 Development Methods

Power Apps - Main communication was through Discord and Zoom. Weekly updates were required and setbacks and roadblock needed to be identified. The use of git and github was not needed since Power Apps is a low code solution that is hosted through MS Office 365 tools. The team was split in two in order to handle both project requirements. Development was modular meaning database tables were reimplemented one by one so that debugging was more manageable.

Enrollment Application - Our development method attempted to mimic agile development. On a weekly basis, our team would have a task assigned to do and subgroups would be assigned a certain functionality to complete. The tasks assigned would be finished within usually one week but sometimes two weeks if there were any problems within the code. Tasks were completed remotely through Github and communications through Discord.

3. Architectural Strategies

Power Apps

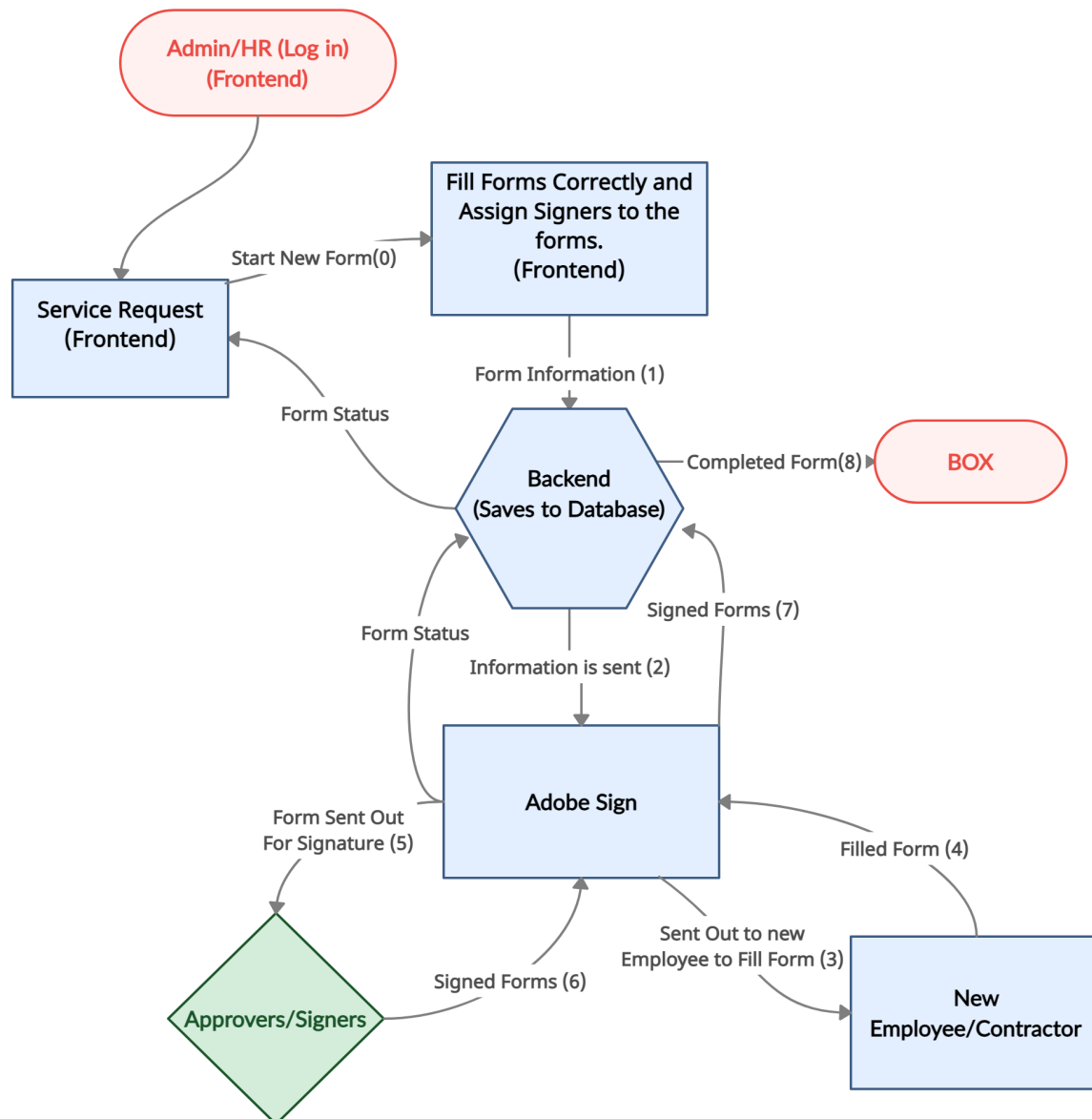
- Power Apps Canvas was used in the creation of both PDGO and Helpdesk apps. The datasource was originally sharepoint lists. Design and migration needed to be implemented to an Azure database.
- Canvas uses prebuilt UI elements that are available through Microsoft's low code solution
- Prebuilt Office 365 functions were implemented to handle user data retrieval
- Microsoft SQL Server Management Studio was used to create new tables, stored procedures, schemas, and to copy table information.
- Possible future plans would be to fully implement power app functions that works natively with managing and updating the SQL database tables

Enrollment Application

- Programming languages that are used are HTML and CSS for the frontend development and Java for the backend development. The use of these languages was because we were building on top of the previous code, and they had used Java, HTML, and CSS.
- The webform was changed from the original web form. The UI our team created is easy to navigate and user-friendly. There is no information cluttering, and the buttons are straightforward that allow the user to go where they want to go.
- SQL Database was used because the previous application made use of it to store information about individual employee/contractor service requests.
- Spring Boot was used for the backend because the previous application made use of it as their backend framework of choice since they had to have CRUD capabilities and REST architecture.

4. System Architecture

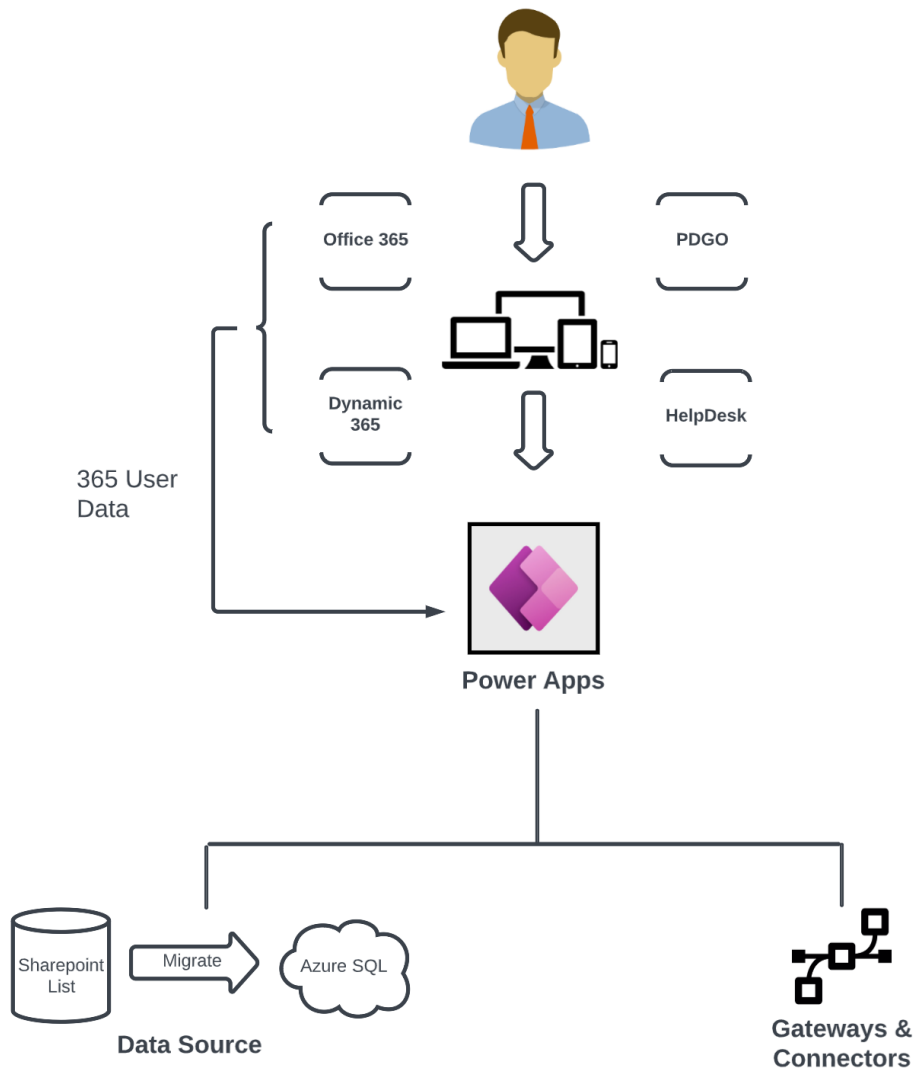
Enrollment Application



- Admin/HR authenticate with username and password to interact with the front-end.
- The service request page is the home page of the Enrollment Application.
- Admin can navigate throughout the app using the navigation bar.
- Admin can start a new form and fill in the “new” employee information to the input fields of the new form.

- Adobe Sign receives the information and sends it out to the new Employee. Employee fills in the rest of the require input fields.
- Adobe Sign receives the forms and sends them to the corresponding approvers/signers. Once all have signed, it returns to Adobe Sign.
- Signed Forms return to the database to be stored.
- Finished documents will be stored in BOX.

Power Apps



- The data source module originally used sharepoint lists for storing data for both PDGO and Helpdesk. The data source was changed to Azure SQL database
- 365 modules include functions and tables that involve user/staff information derived from email data to be used in both Power Apps

5. Policies and Tactics

5.1 Choice of which specific products used

Power Apps - Development of migrating to a cloud solution required a VPN software called Pulse Secure to be allowed to make changes to the Azure cloud database. Microsoft SQL Server Management Studio (SSMS) was used to develop new tables, schemas and procedures for the Power Apps environment. Power Apps canvas was used as the main coding environment.

Enrollment Application - Visual Studio Code was used to change the front-end. When users submit the form, the data is stored in a SQL database. The Spring Boot application was changed in Eclipse EE so that the back-end would communicate with the SQL database and front-end. For the approval workflow, we used what the previous application used, which was Adobe Sign. All of these technologies were used in the last application, so there was no need to change to an alternative.

5.2 Plans for ensuring requirements traceability

Enrollment Application & Power Apps - Our plan to trace the project of requirements involved looking over the emails and text files that we posted within our discord channel for easy accessibility and comparing them to our existing version of our program.

5.3 Plans for testing the software

Enrollment Application - Add test data into the web form and submit the form so that it shows up on the back-end and SQL database. Then send the data to Adobe Sign to start the signing process which should send the form to all necessary users, and then save the document to BOX.

5.4 Plans for maintaining the software

Enrollment Application & Power Apps - Plans to maintain the program will be left to the Public Defender IT team.

5.5 Interfaces for end-users

Enrollment Application - Employees, contractors, and administrators work with the Angular SPA interface. Requestors and approvers interface directly with the Adobe Sign platform.

6. Detailed System Design

6.1 Angular Single Page Application

6.1.1 Responsibilities

Angular is primarily responsible for providing users with a clean interface and smooth process to enter their information. It's also responsible for formatting the user's data with the correct field structure in JSON form to send to the Spring Boot back-end program. It will then generate a signable PDF that will create an efficient method of registering new users into the system. This provides a smoother transition for incorporating new clients into Public Defenders and this will make it much more easier for IT to manage.

6.1.2 Constraints

The Adobe Sign process can only be accessed by an admin. Assumed that the back-end program interacts with the Adobe Sign API every 60 days so that the Refresh Token does not go out of date. The SQL database is assumed to be able to store tuples with many fields. It is also assumed that the backend program will receive data from the front-end program that corresponds to the field structure of the back-end program.

6.1.3 Composition

These REST endpoints are mapped to the appropriate SQL relations: Service Request, Admin, Div Chief / Manager, Dept. Info Security Officer, Application Coordinator. The following SQL relations are for internal use only: Field, Form, Mapping.

6.1.4 Uses/Interactions

Angular contains three subcomponents: HTTP service, Angular router, and the form (user and admin). The HTTP service facilitates communication with the Spring Boot back-end web app. The Angular router enables multiple views and allows navigation between these views. Lastly, the form enables the user to perform data-entry tasks and admins to login and update the user's information and profile.

6.1.5 Resources

An SQL database is used to display and store information from the filling out of the user to generate a PDF.

6.1.6 Interface/Exports

The "model" package is where all the class representations of various objects are contained. The "controller" package contains logic for the public endpoints of the back-end application. Much of the program logic lies here. The "service" package contains a file with logic for sending emails from the application. The "signapi" package contains Adobe Sign credentials and logic to interact with the Adobe Sign API. All HTTP calls to Adobe Sign's REST API are handled here.

6.2 Power Apps Applications

6.2.1 Responsibilities

Power Apps is mainly responsible for the creation and editing of both apps; PDGO & Helpdesk. Both apps connect to an Azure SQL data source that contains various tables related to the respective application. The data sources are also responsible for data entry, retrieval, and posting updated data. PDGO and Helpdesk serve their own functions but the underlying system relies on Power Apps.

6.2.2 Constraints

It is assumed that the user is a staff member of LACPD and has appropriate credentials to access both applications. The user needs to have an outlook email registered with Public Defender in order to have the appropriate data populate within the power apps.

6.2.3 Composition

Power Apps relies heavily on its data sources in order to create the appropriate components that are created within power apps canvas. The data sources include Office365 Users, Email, Power App flows, Notifications, and the Azure database tables.

6.2.4 Uses/Interactions

When a user opens the app, data is gathered from their Office email and is immediately populated within the app. Within login, the email provided decides whether the user is an admin or a staff member. If the user is an admin, they have full access to all modules that are within the power apps. If the user is staff, then they will only have access to modules that staff are only allowed to use. Microsoft platform allows ease of use when it comes to immediately available user data.

6.2.5 Resources

An Azure SQL database with mandatory permissions is used to display and store information from the LACPD power apps..

6.2.6 Interfaces/Exports

Interfaces that are available depend on if the user is staff or admin; determined by Office 365 email.

7. Detailed Lower level Component Design

7.1 SQL Database

7.1.1 Classification

This Component may be classified as a *database*.

7.1.2 Processing Narrative (PSPEC)

N/A

7.1.3 Interface Description

Developers can interface with a SQL database like MySQL or a helper GUI like Adminer.

7.1.4 Processing Detail

7.1.4.1 Design Class Hierarchy

N/A

7.1.4.2 Restrictions/Limitations

The chosen database must be compatible with the .sql configuration file we have created. This involves support for relations with many tuples and fields.

7.1.4.3 Performance Issues

When submitting the individual data entries, there are not many performance issues. If there is a high volume of data being entered into the database, then there might be some delays.

7.1.4.4 Design Constraints

N/A

7.1.4.5 Processing Detail For Each Operation

The backend program makes and changes data on the SQL database using the Hibernate Framework. Developers specify operations in Java and using the Hibernate Framework, it translates the operations into discrete commands in SQL when it interacts with the database.

8. User Interface

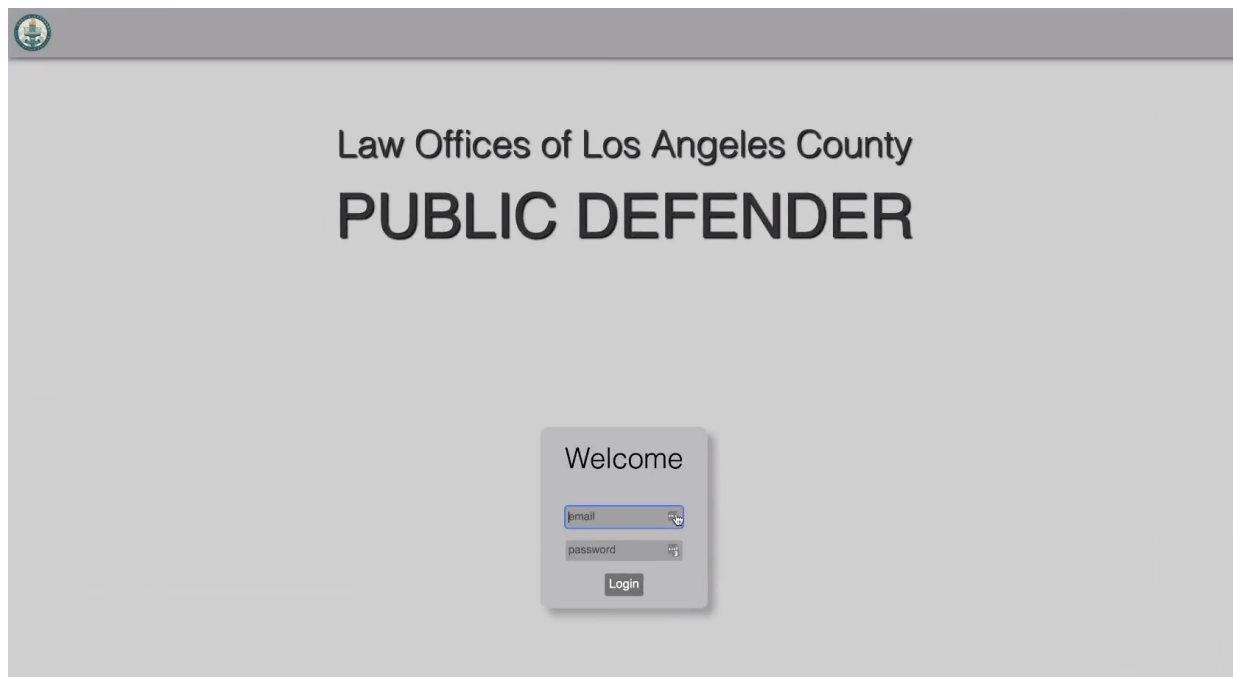
8.1 Overview of User Interface

The UI is a simple design and should be easy to navigate. The admin that logs in to the application at the start will be prompted to the "Service Request" page. This page shows the ongoing service requests and the now usable navigation bar on the top right. The admin can start a new form, and there will be fields that the admin will have to fill out for the new employee. Admin can submit the form to be saved in the database and listed on the service request page for reviewing and eventually sent to Adobe Sign for signatures.

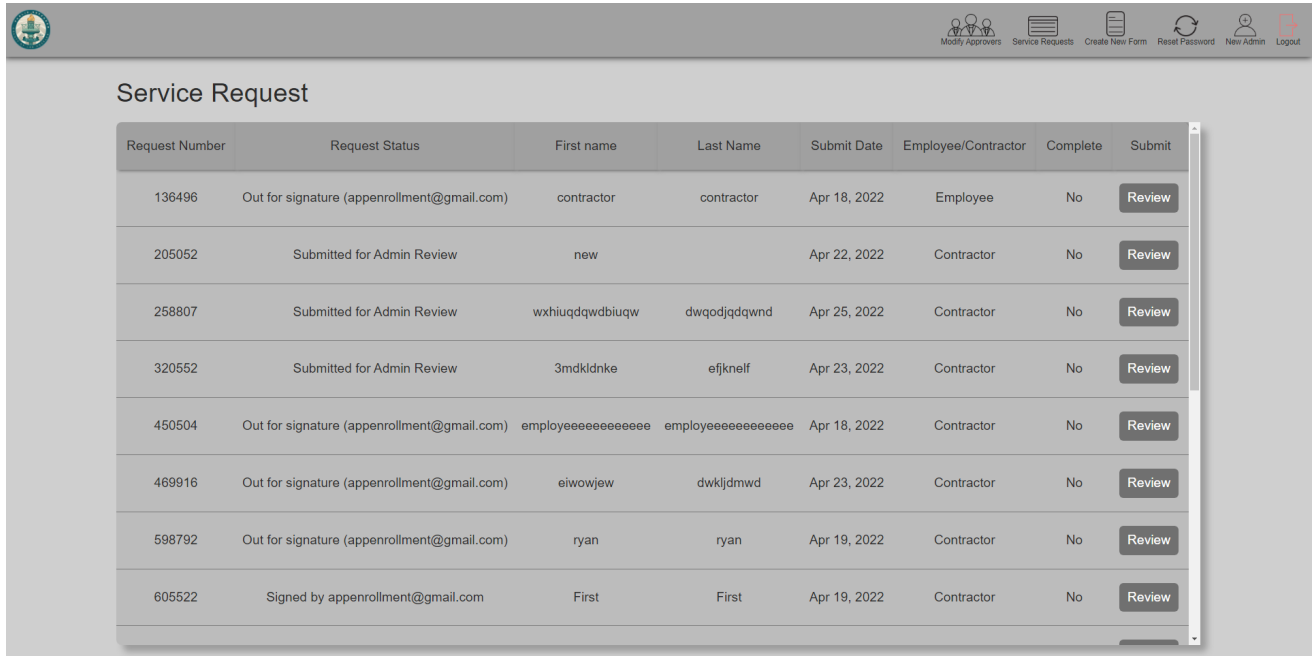
8.2 Screen Frameworks or Images

These can be mockups or actual screenshots of the various UI screens and popups.

1. Login Page



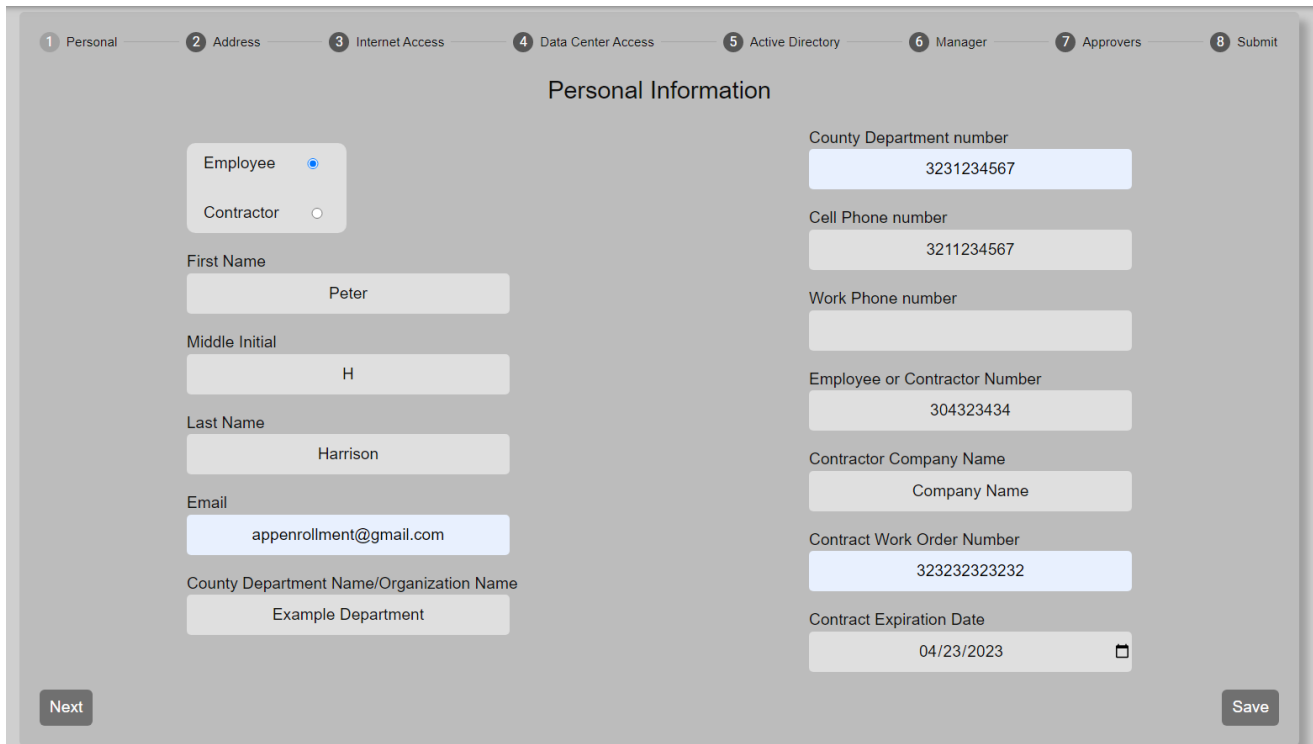
2. Service Request Page



The screenshot shows the 'Service Request' page of a system. At the top, there is a navigation bar with icons for 'Modify Approvers', 'Service Requests', 'Create New Form', 'Reset Password', 'New Admin', and 'Logout'. The main content area is titled 'Service Request' and contains a table with 8 columns: Request Number, Request Status, First name, Last Name, Submit Date, Employee/Contractor, Complete, and Submit. The table lists 8 requests, each with a 'Review' button. The status of the requests varies, including 'Out for signature' and 'Submitted for Admin Review'.

Request Number	Request Status	First name	Last Name	Submit Date	Employee/Contractor	Complete	Submit
136496	Out for signature (appenrollment@gmail.com)	contractor	contractor	Apr 18, 2022	Employee	No	<button>Review</button>
205052	Submitted for Admin Review	new		Apr 22, 2022	Contractor	No	<button>Review</button>
258807	Submitted for Admin Review	wxhiuqdwdbiuqw	dwqodjqdqwnd	Apr 25, 2022	Contractor	No	<button>Review</button>
320552	Submitted for Admin Review	3mdkldnke	efjknelf	Apr 23, 2022	Contractor	No	<button>Review</button>
450504	Out for signature (appenrollment@gmail.com)	employeeeeeeeeee	employeeeeeeeeee	Apr 18, 2022	Contractor	No	<button>Review</button>
469916	Out for signature (appenrollment@gmail.com)	eiwowjew	dwkljdmwd	Apr 23, 2022	Contractor	No	<button>Review</button>
598792	Out for signature (appenrollment@gmail.com)	ryan	ryan	Apr 19, 2022	Contractor	No	<button>Review</button>
605522	Signed by appenrollment@gmail.com	First	First	Apr 19, 2022	Contractor	No	<button>Review</button>

3. Start a new Form



The screenshot shows the 'Start a new Form' page, which is a multi-step process. The steps are: 1 Personal, 2 Address, 3 Internet Access, 4 Data Center Access, 5 Active Directory, 6 Manager, 7 Approvers, and 8 Submit. The current step is 'Personal Information'. The form contains several input fields for personal and contact information, including name, email, and phone numbers. There are also radio buttons for 'Employee' and 'Contractor'. The form is designed to be user-friendly with clear labels and a logical flow.

1 Personal 2 Address 3 Internet Access 4 Data Center Access 5 Active Directory 6 Manager 7 Approvers 8 Submit

Personal Information

☒ Employee ☐ Contractor

First Name:

Middle Initial:

Last Name:

Email:

County Department Name/Organization Name:

County Department number:

Cell Phone number:

Work Phone number:

Employee or Contractor Number:

Contractor Company Name:

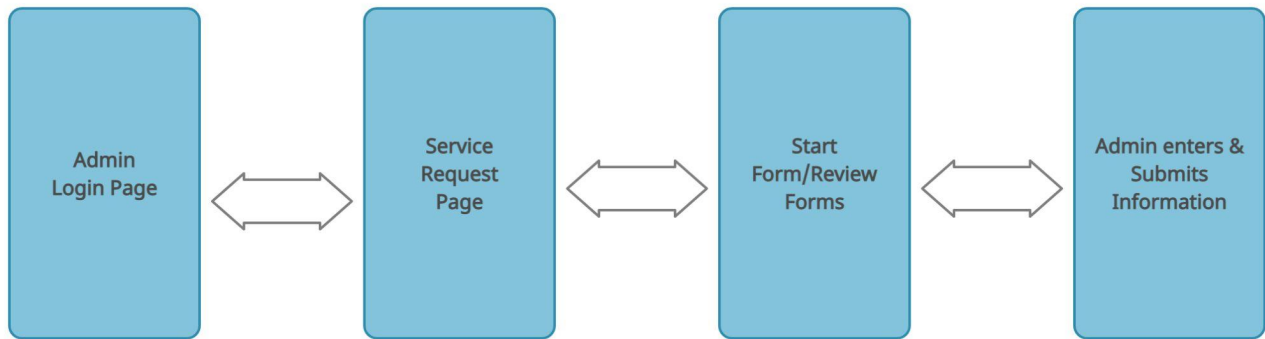
Contract Work Order Number:

Contract Expiration Date:

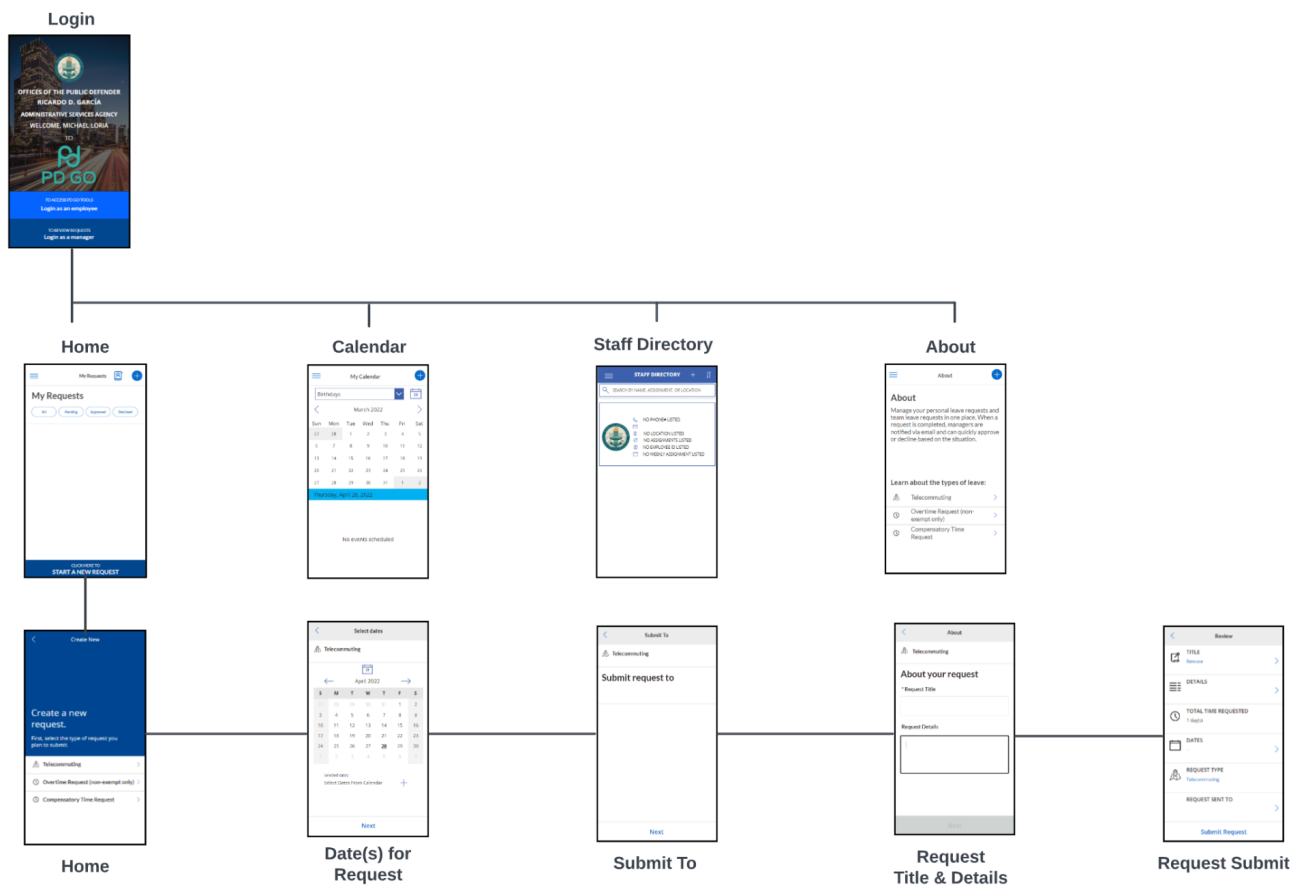
Next Save

8.3 User Interface Flow Model

The flow model demonstrates the front-end admin user interface.



Power Apps (PDGO)



9. Database Design

Our back-end Spring Boot program makes use of an SQL database. Multiple REST endpoints are associated with various tables in the database. The relations are as follows:

Public (REST Endpoints): Service Request, Admin, Div Chief / Manager, Dept. Info Security Officer, Application Coordinator

Private (Internal): Mappings, Forms, Fields

Public Endpoints

1. Service Requests

METHOD	ENDPOINT	DESCRIPTION
GET	service_requests	Gets all the service requests in the database
GET	service_requests/{requestNumber}	Get a specific service request based on requestNumber
POST	service_requests	Add a service request
PATCH	service_requests/{requestNumber}	Update a service request
DELETE	service_requests/{requestNumber}	Delete a service request

2. Request Statuses

METHOD	ENDPOINT	DESCRIPTION
GET	request_statuses/{id}	Get the service request's request status

3. Admin-related Endpoints

METHOD	ENDPOINT	DESCRIPTION
GET	admin/service_requests	Gets all the service requests (condensed to 4 fields) in the database
GET	admin/service_requests/{requestNumber}	Get a specific service request based on requestNumber
PATCH	admin/service_requests/{requestNumber}	Make completely unrestricted updates to any service request fields
PATCH	admin/reset_password	Update password with new-password header
GET	div_chief_managers	Get all Div Chief Managers
GET	div_chief_managers/{id}	Get a Div Chief Manager based on id
POST	div_chief_managers	Add a new Div Chief Manager

PATCH	department_heads/{id}	Update a Department Head
DELETE	department_heads/{id}	Delete a Department Head
GET	application_coordinators	Get all Application Coordinators
GET	application_coordinators/{id}	Get an Application Coordinator based on id
POST	application_coordinators	Add an new Application Coordinator
PATCH	application_coordinators/{id}	Update an Application Coordinator
DELETE	application_coordinators/{id}	Delete an Application Coordinator

10. Requirements Validation and Verification

Requirement	Satisfied By	Testing Method
Gather relevant employee/contractor onboarding data.	Angular web form (front-end).	Compare PDF fields to web form fields.

Store user data from the web form.	Backend SQL Database.	Run both programs, complete sample form, check SQL database
Allow PD members to review individual service requests.	Admin authentication, notifications, and request review page (Angular front-end).	Login to “Admin” page of Angular front-end, access the “Review Request” page of any particular request.
Allow approvers to be selected for a particular request.	Approvers page for Admin users.	Add a sample approver, select the approver to use for a sample request, and see if they are notified at their turn in the signing process.
Implement a signing process for the requestor and various approvers.	Integration of Adobe Sign API with the back-end program.	Simply submit a request to Adobe Sign as an Admin user and ensure that all parties involved receive Adobe Sign notifications.

11. Glossary

AngularJS - JavaScript-based open-source front-end web framework used in developing single-page applications

Azure - a public cloud computing platform—with solutions including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) that can be used for services such as analytics, virtual computing, storage, networking, and much more.

IT - Information Technology

Microsoft Power Apps - a suite of apps, services, and connectors, as well as a data platform, that provides a rapid development environment to build custom apps for your business needs

PDF - Portable Document Format

REST API - Representational state transfer is a software architectural style that uses a subset of HTTP. It is commonly used to create interactive applications that use Web services. A Web service that follows these guidelines is called RESTful

Sharepoint - a website-based collaboration system that uses workflow applications, “list” databases, and other web parts and elements

SPA - Single-page applications are web applications that load an HTML page that updates dynamically as the user interacts if the application

SRS - Software Requirements Specification

UI - User Interface

12. References

Software Requirements Specification Cloud Database Migration & LACPD New Hire Enrollment Web Application
Version 1.4 approved

AngularJS Tutorial: <https://www.w3schools.com/angular/>

Power Apps Documentation: <https://docs.microsoft.com/en-us/power-apps/>

Law Offices of Los Angeles County Public Defender Website: <https://pubdef.lacounty.gov/>

Adobe Sign REST API Documentation <https://www.adobe.io/apis/documentcloud/sign/docs.html>

Spring Boot REST API Tutorial <https://spring.io/guides/tutorials/rest/>

CSNS Project Page: <https://csns.cysun.org/departments/cs/project/view?id=7808904>